Computers & Operations Research 39 (2012) 74–85

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Three-dimensional container loading models with cargo stability and load bearing constraints

Leonardo Junqueira, Reinaldo Morabito*, Denise Sato Yamashita

Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Brazil

ARTICLE INFO

Available online 27 July 2010

Keywords:
Cutting and packing problems
Three-dimensional container loading
Combinatorial optimization
Mathematical modeling

ABSTRACT

Mathematical models for the problem of loading rectangular boxes into containers, trucks or railway cars have been proposed in the literature, however, there is a lack of studies which consider realistic constraints that often arise in practice. In this paper, we present mixed integer linear programming models for the container loading problem that consider the vertical and horizontal stability of the cargo and the load bearing strength of the cargo (including fragility). The models can also be used for loading rectangular boxes on pallets where the boxes do not need to be arranged in horizontal layers on the pallet. A comprehensive performance analysis using optimization software with 100s of randomly generated instances is presented. The computational results validate the models and show that they are able to handle only problems of a moderate size. However, these models might be useful to motivate future research exploring other solution approaches to solve this problem, such as decomposition methods, relaxation methods, heuristics, among others.

1. Introduction

The three-dimensional container loading problem, in its basic form, consists of finding the best three-dimensional packing pattern for loading a set of rectangular boxes into a container so that the total volume (or value) of the boxes loaded is maximized, and the boxes do not overlap (e.g., [21,20,38,37,35,43,41,49,25,42]). The problem considered here can be broadly characterized as being of type 3/B/O/-, according to the typology presented in [14], or as being of a type 3D-R-IIPP/SLOPP/SKP, according to the typology presented in [50]. In other words, it is assumed that a consignment of goods wrapped up in boxes is to be loaded into a single container of known dimensions, and that boxes and containers are assumed to be of rectangular shape.

Other considerations besides the non-overlapping constraints often arise in practice like vertical and horizontal stability of the cargo, load bearing strength and fragility of the cargo, grouping or separation of items inside a container, multi-drop situations, complete shipment of certain item groups, container weight limit, weight distribution within a container, among others (e.g., [5,46,27]). These assumptions have motivated the study and development of various approximate methods (heuristics and metaheuristics) in the literature to solve the container loading problem (e.g., [17,7,26,34,39,51,18]). However, not much work has presented mathematical models for these problems with additional constraints. In the present paper, we are particularly interested in the mathematical formulation of the problem with cargo stability and load bearing constraints. To the best of our knowledge, we are not aware of other papers that have presented mathematical formulations which explicitly deal with such constraints.

Cargo stability refers to the support of the bottom faces of boxes, in the case of vertical stability (i.e., the boxes must have their bottom faces supported by other box top faces or the container floor), and the support of the lateral faces of boxes, in the case of horizontal stability (e.g., [5,45,13,47,6,15,17,43,7,16,46,26,34,39,51,18,31]). Load bearing strength refers to the maximum number of boxes that can be stacked one above each other, or more generally, to the maximum pressure that can be applied over the top face of a box, so as to avoid damaging the box (e.g., [45,44,6,17,4,18,31,11]). We note that fragility is the particular case of load bearing where boxes cannot be placed above a fragile box, since its top face does not bear any kind of pressure. The practical importance of incorporating these groups of constraints to the problem is to avoid loading patterns where boxes are "floating in mid-air" inside a container, or where products are damaged due to deformation of the boxes that contain them [5,46,4].

The 0–1 integer linear programming models presented in this study can also be applied to dealing with the three-dimensional problem of loading rectangular boxes on pallets, where the boxes do not need to be arranged in horizontal layers [24,19,33,3]. Moreover, the models can also be extended to cope with other

* Corresponding author.
E-mail addresses: leo_junqueira@yahoo.com (L. Junqueira), morabito@ufscar.br (R. Morabito), denisesy@dep.ufscar.br (D. Sato Yamashita).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.07.017

practical constraints, as discussed in the next sections. To evaluate the performance of the models, they were coded in the modeling language GAMS and the CPLEX solver was used to solve different randomly generated problem instances. This work is organized as follows. In Section 2, we briefly describe the three-dimensional container loading problem and some practical constraints that may arise when solving this problem. In Section 3, we present the mathematical models for the problem with cargo stability and load bearing constraints. In Section 4, we analyze the results of some computational tests for the proposed models solved by GAMS/CPLEX. Finally, in Section 5, we present concluding remarks and some perspectives for future research.

2. Problem definition

The problem of loading boxes into containers can be classified in four variants [43]: the *Strip Packing Problem* (SPP), the *Bin Packing Problem* (BPP), the *Multi-Container Loading Problem* (MCLP), and the *Knapsack Loading Problem* (KLP). The SPP considers a container where two of its dimensions are fixed (e.g., width and height) and the third dimension (e.g., length) is a variable of the problem. Then, the problem consists of deciding how to load all boxes of different sizes inside the container, so that the variable dimension (length) is minimized (e.g., [28,37,8]). In the BPP, there are multiple containers (or bins) of the same fixed sizes and costs, and the problem consists of deciding how to load all boxes, so that the total number of used containers is minimized (e.g., [35,47,46,18]). Unlike the BPP, in the MCLP the containers do not necessarily have the same sizes and costs, and the problem consists of deciding how to load all the boxes so that the total cost of the chosen subset of containers to be loaded is minimized (e.g., [9,16]).

In the present paper, we are particularly concerned with the KLP, where there is only one container of a fixed size, each box has an associated size and value, and the objective is to load a subset of boxes inside the container so that the total value of the loaded boxes is maximized. Obviously if each box value is proportional to the volume of the box, then the problem consists of maximizing the total volume of the loaded boxes or, equivalently, minimizing the empty spaces left in the container. Nevertheless, it is worth mentioning that the models presented in the next sections can be straightforwardly extended to also deal with other variants of the container loading, as the SPP and BPP.

The KLP was studied by various authors (e.g., [20,38,5,44,13,35,15,17,23,32,43,7,29,10,26,34,30,39,51,4,49,11,25]). It can also be seen as a three-dimensional problem of loading rectangular boxes onto pallets. The problem of carrying boxes on pallets can be separated into two cases [24]: the *Manufacturer's Pallet Loading Problem* (MPLP) and the *Distributor's Pallet Loading Problem* (DPLP). In the first case (MPLP) there is only one type of box (i.e., all boxes are of the same size), while in the DPLP there is more than one type. Both problems, the MPLP and the DPLP can be solved in their two- or three-dimensional version, although the first one is more common in practice. The difference between them is that in the two-dimensional case, the loading pattern is built in horizontal layers on the pallet, while in the three-dimensional one, the loading pattern can be generic. The last case can also be seen as a KLP with only one type of box [38,5].

Twelve practical considerations that might play an important role when modeling more realistic container loading problems were presented in [5]. Besides cargo stability and load bearing, this list of considerations also includes: box orientation, box handling, box grouping, box separation, complete shipment of box groups, box priorities, complexity of the arrangement, container weight limit, weight distribution within the container, multi-dropping, among others. Box orientation means that some boxes must be loaded inside the container in a pre-determined orientation. Box handling means that some boxes, due to their dimension, or weight, or even the equipment used to load/unload the shipment, must be placed in pre-determined positions inside the container. Box grouping means that boxes with the same destination, or of the same type, must be placed close to each other in the container (e.g., [5,47,16]). Box separation means that boxes that cannot come into contact should be placed far from each other in the container (e.g., [16]).

Complete shipment of box groups means that boxes that constitute a functional entity (e.g., components for assembly into a piece of machinery) must be shipped in the same container (e.g., [47]). Box priorities means that boxes with products with a closer due date or expiration date, for example, may have greater priority than others. Complexity of the arrangement means that boxes placed in complex loading patterns may result in a greater material handling effort due to, for example, the limitation of the equipment used to load/unload the shipment. Weight limit of the container means that heavy boxes must be loaded into the container without exceeding the maximum weight limit supported by the container (e.g., [45,47,6,17,18]). Weight distribution within the container means that the center of gravity of the container should be close to the geometrical mid-point of the container floor (e.g., [13,6,15,16]). Multi-drop constraints refer to cases where boxes that are delivered to the same destinations must be placed close to each other in the container (or truck, as it is more common in practice), the loading patterns must take into account the delivery route of the vehicle and the order in which the boxes are unloaded (e.g., [5,45,47,26,39,18,31,11,40]).

Although the aforementioned studies consider the practical issues described, in general, they do not present mathematical formulations for them. Some papers, such as [1,22,36,2], present formulations for two-dimensional cutting and packing problems that can be easily extended to the three-dimensional KLP. Other formulations for the KLP are presented in [48] and [9]. Nevertheless, these papers do not handle the practical considerations mentioned, limiting only to avoid that the boxes do not overlap inside a container. As mentioned, the KLP models of the next section consider cargo stability and load bearing constraints, but they can also be easily modified to handle other practical constraints such as box orientation, box handling, complete shipment of box groups, box priorities and container weight limit. For instance, the weight limit consideration can be modeled as a linear knapsack constraint (i.e., the sum of the weights of all boxes loaded inside the container must be smaller or equal to the weight limit supported by the container).

3. Mathematical formulations

We assume there are different types of boxes with given length l_i, width w_i, height h_i, value v_i, and a maximum quantity b_i, $i=1,\ldots,m$, which can be loaded inside the object (container, truck, railroad car or pallet) with given length L, width W and height H (when considering a pallet, H is the maximum allowed height of the cargo loading). The dimensions of the boxes are integer, and they can only be placed orthogonally into the container (i.e., the edges of a box are either parallel or perpendicular to the axes of the container), and their orientation is fixed (i.e., the boxes cannot rotate). This last assumption can be easily relaxed in the models presented and here it is considered only to simplify the presentation of the formulations.

A Cartesian coordinate system is adopted with its origin in the container's front-left-bottom corner, and let (x, y, z) be the possible coordinates where the front-left-bottom corner of a box can be placed. These possible positions along axes L, W and H of the container belong to the sets: $X=\{0,1,2,\ldots,L-\min_i(l_i)\}$,

$Y=\{0,1,2,\ldots,W-\min_i(w_i)\}$ and $Z=\{0,1,2,\ldots,H-\min_i(h_i)\}$, respectively. As pointed out in [12,1], for a given cutting or packing pattern, each packed box could be moved down and/or forward and/or to the left, until its bottom, front and left-hand face are adjacent to other boxes or to the container. These patterns, called normal patterns, allowed us, without loss of generality, to restrict the sets X, Y and Z to

$$X=\{x|x=\sum_{i=1}^{m}\varepsilon_i\cdot l_i, 0\le x\le L-\min_i(l_i),\ 0\le\varepsilon_i\le b_i \text{ and integer}, i=1,\ldots,m\}, \quad (1)$$

$$Y=\{y|y=\sum_{i=1}^{m}\varepsilon_i\cdot w_i,\ 0\le y\le W-\min_i(w_i),\ 0\le\varepsilon_i\le b_i \text{ and integer}, i=1,\ldots,m\}, \quad (2)$$

$$Z=\{z|z=\sum_{i=1}^{m}\varepsilon_i\cdot h_i,\ 0\le z\le H-\min_i(h_i),\ 0\le\varepsilon_i\le b_i \text{ and integer}, i=1,\ldots,m\}. \quad (3)$$

Fig. 1 shows a possible placement of a box of type i inside the container. In order to describe the constraints that avoid overlapping of boxes inside the container, let $c_{ixyzx'y'z'}$, $i=1,\ldots,m$, x, $x'\in X$, y, $y'\in Y$ and z, $z'\in Z$, be defined as

$$c_{ixyzx'y'z'}=\begin{cases}1 & \text{if a box of type } i \text{ placed with its front-left-bottom corner at } (x,y,z)\text{, occupies point } (x',y',z');\\ 0 & \text{otherwise.}\end{cases}$$

The mapping $c_{ixyzx'y'z'}$ is not a decision variable and it is computed *a priori* as follows:

$$c_{ixyzx'y'z'}=\begin{cases}1 & \text{if } 0\le x\le x'\le x+l_i-1\le L-1;\\ & 0\le y\le y'\le y+w_i-1\le W-1;\\ & 0\le z\le z'\le z+h_i-1\le H-1;\\ 0 & \text{otherwise.}\end{cases}$$

Let $X_i=\{x\in X|0\le x\le L-l_i\}$, $Y_i=\{y\in Y|0\le y\le W-w_i\}$ and $Z_i=\{z\in Z|0\le z\le H-h_i\}$, $i=1,\ldots,m$. The decision variables a_{ixyz}, $i=1,\ldots,m$, $x\in X_i$, $y\in Y_i$, $z\in Z_i$, of the model are defined as

$$a_{ixyz}=\begin{cases}1 & \text{if a box of type } i \text{ is placed with its front-left-bottom corner at position } (x,y,z)\text{, so that}\\ & 0\le x\le L-l_i,\quad 0\le y\le W-w_i \text{ and } 0\le z\le H-h_i;\\ 0 & \text{otherwise.}\end{cases}$$

The problem of loading boxes inside a single container, without additional considerations of cargo stability or load bearing constraints, can be written as a direct extension of a 0-1 integer linear programming model proposed in [1] for the two-dimensional non-guillotine cutting problem:

$$\max \sum_{i=1}^{m}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} v_i\cdot a_{ixyz}, \quad (4)$$

$$\sum_{i=1}^{m}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} c_{ixyzx'y'z'}\cdot a_{ixyz}\le 1,\quad x'\in X,\ y'\in Y,\ z'\in Z, \quad (5)$$

$$\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} a_{ixyz}\le b_i,\quad i=1,\ldots,m, \quad (6)$$

$$a_{ixyz}\in\{0,1\}\quad \begin{array}{l} i=1,\ldots,m,\\ x\in X_i,\quad y\in Y_i,\ z\in Z_i.\end{array} \quad (7)$$

In formulation (4)–(7), the objective function (4) aims to maximize the total value of the boxes packed inside the container (if $v_i=(l_i.w_i.h_i)$, then (4) maximizes the total volume of the boxes), constraints (5) avoid the overlapping of the boxes packed, constraints (6) limit the maximum number of boxes packed, and constraints (7) define the domain of the decision variables. In order to save computer memory when compiling model (4)–(7) in a modeling language such as GAMS, the mapping $c_{ixyzx'y'z'}$ does not need to be pre-computed explicitly in (5); instead, (5) can be rewritten as

$$\sum_{i=1}^{m}\sum_{\{x\in X_i|x'-l_i+1\le x\le x'\}}\sum_{\{y\in Y_i|y'-w_i+1\le y\le y'\}}\sum_{\{z\in Z_i|z'-h_i+1\le z\le z'\}} a_{ixyz}\le 1,\quad x'\in X,\ y'\in Y,\ z'\in Z. \quad (8)$$

For the remaining part of this paper, we will use the term *base model* to refer to the model comprised by (4), (6)–(8). Note that, in the computational experiments of Section 4, it was not possible to compile some of the models due to the computer memory required to pre-compute $c_{ixyzx'y'z'}$; however, these same models could be compiled and solved with expression (8). It is also worth noting that the base model can be easily extended to consider special cases of (i) rotation of the boxes, (ii) packing boxes into multiple containers, and (iii) packing boxes into containers with inner obstacles ("defects"), as already discussed in [1] for the two-dimensional non-guillotine cutting problem.

3.1. Cargo vertical and horizontal stability constraints

We consider two kinds of loading stability: vertical and horizontal. *Vertical* stability is related to the capacity of the loaded boxes to withstand the gravity force acceleration over them, i.e., they are not displaced with respect to the z-axis. This kind of stability is also called *static* stability, since it deals with situations where the container or truck is not on the move. *Horizontal* stability, on the other hand, is related to the capacity of the loaded boxes to withstand the inertia of its own bodies, i.e., they are not displaced with respect to the x and y axes. This kind of stability is also known as *dynamic* stability, since it comprehends situations where the container or truck is displaced horizontally, and it is subjected to variations in the speed of the displacement. Since the formulations for both cases are analogous [27], in the sequel we detail the formulation for the vertical stability and then we briefly present it for the horizontal stability.

Vertical stability deals with the stability along the z-axis, since it refers to the support of the bottom face of each box. Fig. 2 (left) shows two boxes positioned inside a container in an unstable vertical way. However, it is possible to rearrange these boxes in such a way to obtain a stable vertical loading, as in Fig. 2 (right).

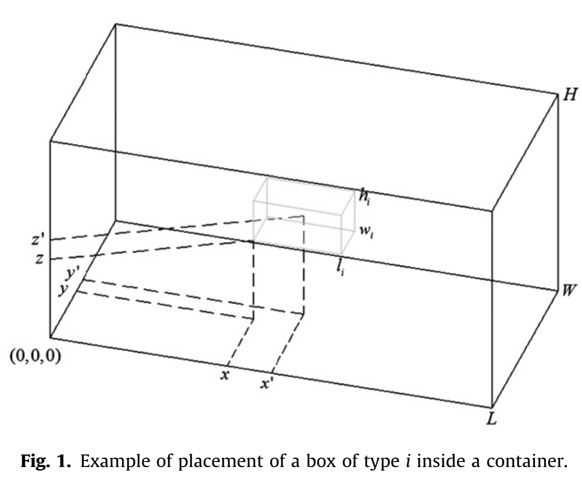

Fig. 1. Example of placement of a box of type i inside a container.

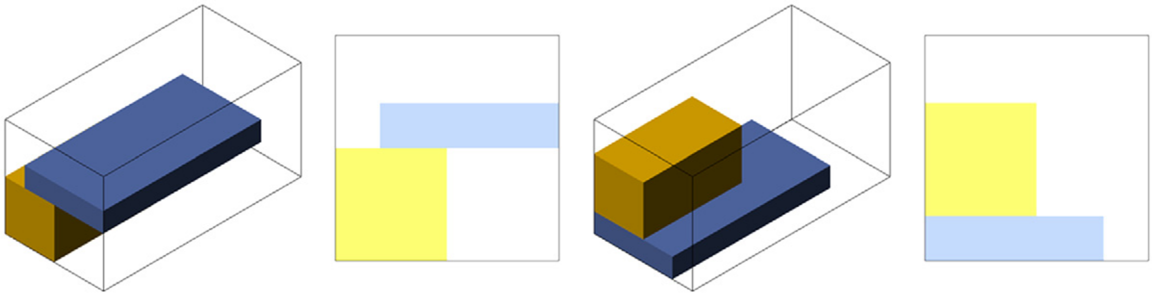

Fig. 2. Two boxes positioned in an unstable (left) and in a stable (right) vertical way, respectively.

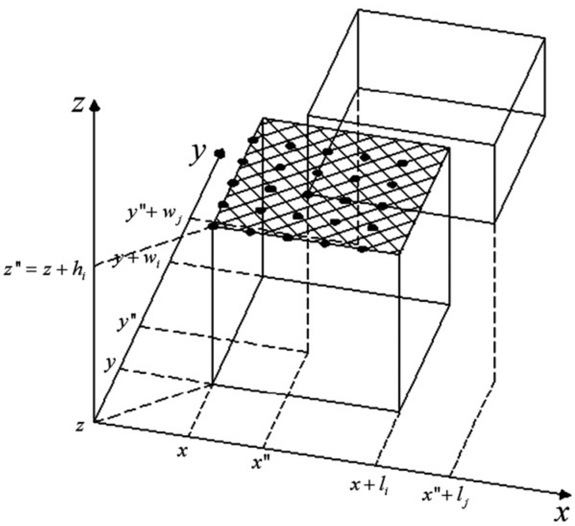

Fig. 3. Relative position of two boxes inside of a container (vertical stability).

Therefore, it is possible to map a set of points that a certain box "offers" to support other boxes placed immediately above its top face by defining the function $d_{ixyz(z+h_i)}$, $i=1,\dots,m$, $x,x' \in X$, $y,y' \in Y$ and $z \in Z$ as:

$$d_{ixyzx'y'(z+h_i)} = \begin{cases} 1 & \text{if a box of type } i \text{ placed with its} \\ & \text{front-left-bottom corner at } (x,y,z)\text{, contains point} \\ & (x',y',z+h_i) \text{ on its top face;} \\ 0 & \text{otherwise.} \end{cases}$$

Note that the subindex $z+h_i$ in $d_{ixyzx'y'(z+h_i)}$ is redundant, however, it was left in the function for notation convenience. Similarly to $c_{ixyzx'y'z'}$, $d_{ixyzx'y'(z+h_i)}$ is not a decision variable of the model, and it is also computed *a priori* as follows:

$$d_{ixyzx'y'(z+h_i)} = \begin{cases} 1 & \text{if } 0 \le x \le x' \le x+l_i-1 \le L-1; \\ & 0 \le y \le y' \le y+w_i-1 \le W-1; \\ & 0 \le z \le H-h_i; \\ 0 & \text{otherwise} \end{cases}$$

Note that $d_{ixyzx'y'(z+h_i)}$ is defined analogously to $c_{ixyzx'y'z'}$, with the difference that the first will map only the set of points at the top face of a box of type i placed at (x, y, z) (except for the points on the back and right-hand edge of the box), and the second will map the set of points inside a box of type i placed at (x, y, z) (except for the points on the top, back and right-hand face of the box). Fig. 3 shows the mapping for these sets of points for two boxes placed inside the container, where the bottom face of the smaller box is partially supported by the top face of the larger box. Note that this mapping determines, for each box of type i placed at (x, y, z), all possible points (x', y', $z+h_i$) that this box can "offer" to support any other boxes of type j (including $j=i$), placed at (x'', y'', z''), with $z''=z+h_i$, i.e., the boxes are in contact (Fig. 3).

Another parameter used in the model is $\alpha_i \in [0,1]$, called parameter of stability of box of type i in the z-axis (or parameter of vertical stability of box of type i). This kind of parameter has already been used in the literature of container loading (e.g., [15]) and it indicates the desired amount of vertical stability for all boxes of type i (in some cases we may have $\alpha_i=\alpha$ for all i). If $\alpha_i=1$, it means that the bottom faces of all boxes of type i loaded must be 100% supported by the top faces of one or more boxes placed immediately below them. If $\alpha_i=0$, it means that there is no requirement of stability for the boxes of type i with respect to the z-axis (for example, the boxes can be only partially supported, or they can be even "floating" inside the container). Considering X, Y and Z defined as the full sets $X=\{0,1,2,\dots,L-\min_i(l_i)\}$, $Y=\{0,1,2,\dots,W-\min_i(w_i)\}$ and $Z=\{0,1,2,\dots,H-\min_i(h_i)\}$ (instead of the reduced sets in (1)–(3)), respectively, then the vertical stability constraint can be stated as:

$$\sum_{\{i=1,\dots,m|z'-h_i \ge 0\}} \sum_{x \in X_i} \sum_{y \in Y_i} \sum_{\{x' \in X|x' \le x' \le x'+l_j-1\}} \sum_{\{y' \in Y|y' \le y' \le y'+w_j-1\}} d_{ixy(z'-h_i)x'y'z'} \cdot a_{ixy(z'-h_i)} \ge \alpha_j \cdot l_j \cdot w_j \cdot a_{jx'y'z'} \quad j=1,\dots,m, \quad x' \in X_j, \quad y' \in Y_j, \quad z' \in Z_j/\{0\}, \tag{9}$$

i.e., a minimum fraction α_j of points at the bottom face of a box of type j must be supported by the points at the top faces of boxes of type i, which are placed immediately below the box of type j (including $j=i$), so that they are in contact (Fig. 3). Note in (9) where $z' \in Z_j/\{0\}$, instead of $z' \in Z_j$, because a box placed on the floor of the container (i.e., at $z''=0$) is always vertically stable. We remark that constraint (9) is not valid if sets X, Y and Z are reduced as in (1)–(3), respectively (normal patterns), since this definition of the sets may generate "holes" in the top face of the box of type i. Analogously to (5), expression (9) can also be rewritten defining function $d_{ixyzx'y'(z+h_i)}$ implicitly:

$$\sum_{\{i=1,\dots,m|z'-h_i \ge 0\}} \sum_{\{x \in X_i|x'-l_i+1 \le x \le x'+l_j-1\}} \sum_{\{y \in Y_i|y'-w_i+1 \le y \le y'+w_j-1\}} \sum_{x'=\max(x,x')}^{\min(x+l_i-1,x'+l_j-1)} \sum_{y'=\max(y,y')}^{\min(y+w_i-1,y'+w_j-1)} a_{ixy(z'-h_i)} \ge \alpha_j \cdot l_j \cdot w_j \cdot a_{jx'y'z'} \quad j=1,\dots,m, \quad x' \in X_j, \quad y' \in Y_j, \quad z' \in Z_j/\{0\}, \tag{10}$$

which can be rewritten as

$$\sum_{\{i=1,\dots,m|z'-h_i \ge 0\}} \sum_{\{x \in X_i|x'-l_i+1 \le x \le x'+l_j-1\}} \sum_{\{y \in Y_i|y'-w_i+1 \le y \le y'+w_j-1\}} L_{ij}^{[1]} \cdot W_{ij}^{[1]} \cdot a_{ixy(z'-h_i)} \ge \alpha_j \cdot l_j \cdot w_j \cdot a_{jx'y'z'}$$

$$\text{where}\begin{cases} L_{ij}^{[1]} = \min(x+l_i, x'+l_j) - \max(x,x'), & j=1,\dots,m, \\ W_{ij}^{[1]} = \min(y+w_i, y'+w_j) - \max(y,y'), & x' \in X_j,\ y' \in Y_j,\ z' \in Z_j/\{0\}, \end{cases} \tag{11}$$

that is, a minimum fraction α_j of the area of the bottom face of a box of type j must be supported by the area of the top faces of boxes of type i placed immediately below and in contact with the box of type j. Fig. 4 shows the area that a box of type i "offers" to support a box of type j. Note that, differently from (9), constraints (10) and (11) are valid if sets X, Y and Z are defined as in (1)–(3), since they consider all possible "holes" at the top face of box of type i, contributing with all points $(x', y', z+h_i)$.

However, constraints (10) and (11) with sets defined as in (1)–(3) are valid only for $\alpha_i = 1$. Fig. 5 shows an example of this situation for a container with size $(L,W,H)=(12,5,2)$ and two types of boxes $(l_1,w_1,h_1)=(6,3,1)$ with $b_1=2$, and $(l_2,w_2,h_2)=(10,5,1)$ with $b_2=1$. Note that there is loss of generality when considering the *base model* with constraint (11) using (1)–(3) with $\alpha_1=0.8$. This pattern is worse than the optimal pattern, obtained considering the *base model* with constraint (11) using the original full sets X, Y and Z with $\alpha_1=0.8$. The use of (1)–(3) excludes position $(x, y, z)=(1, 0, 0)$ that is used by the box of type 2.

As mentioned above, the formulations for horizontal stability can be deduced similarly to the formulations for vertical stability. Horizontal stability deals with the stability along the x and y axes, since it refers to the support of the left and the front faces of each box, respectively. Therefore, analogously to parameter α_i for the vertical stability, we can define parameters $\beta_i \in [0,1]$ and $\gamma_i \in [0,1]$, for the stability of box of type i in the x- and y-axis, respectively. That is, a minimum fraction $\beta_i(\gamma_i)$ of points (or the area) at the left (front) face of a box of type i must be supported by the points (or the area) at the right (back) face of boxes of type j, which are placed immediately at the left (in front) of box of type i (including $i=j$), so that they are in contact. In particular, constraints (12) and (13) presented below are the equivalent ones for constraint (11) for vertical stability:

$$\sum_{\{i=1,\dots,m \mid x'-l_i \ge 0\}} \sum_{\{y \in Y_i \mid y'-w_i+1 \le y \le y'+w_j-1\}} \sum_{\{z \in Z_i \mid z'-h_i+1 \le z \le z'+h_j-1\}} W_{ij}^{[1]} \cdot H_{ij}^{[1]} \cdot a_{i(x'-l_i)yz} \ge \beta_j \cdot w_j \cdot h_j \cdot a_{jx'y'z'}$$

$$\text{where}\begin{cases} W_{ij}^{[1]} = \min(y+w_i, y'+w_j) - \max(y,y'), & j=1,\dots,m, \\ H_{ij}^{[1]} = \min(z+h_i, z'+h_j) - \max(z,z'), & x' \in X_j/\{0\} \quad y' \in Y_j,\ z' \in Z_j, \end{cases} \tag{12}$$

$$\sum_{\{i=1,\dots,m \mid y'-w_i \ge 0\}} \sum_{\{x \in X_i \mid x'-l_i+1 \le x \le x'+l_j-1\}} \sum_{\{z \in Z_i \mid z'-h_i+1 \le z \le z'+h_j-1\}} L_{ij}^{[1]} \cdot H_{ij}^{[1]} \cdot a_{ix(y'-w_i)z} \ge \gamma_j \cdot l_j \cdot h_j \cdot a_{jx'y'z'}$$

$$\text{where}\begin{cases} L_{ij}^{[1]} = \min(x+l_i, x'+l_j) - \max(x,x'), & j=1,\dots,m, \\ H_{ij}^{[1]} = \min(z+h_i, z'+h_j) - \max(z,z'), & x' \in X_j,\quad y' \in Y_j/\{0\},\ z' \in Z_j. \end{cases} \tag{13}$$

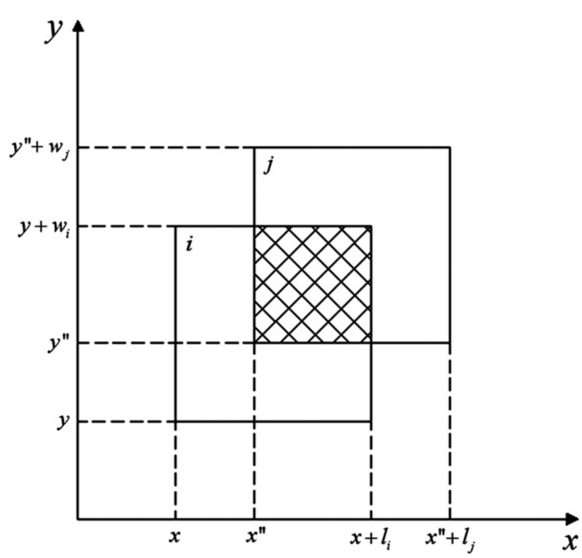

Fig. 4. Contact area between two boxes (vertical stability).

3.2. Load bearing and fragility constraints

In order to consider the load bearing strength of the cargo, a new constraint should be added to the model. In this case, this new constraint can limit the maximum number of boxes placed above a box of type i (not necessarily in direct contact to it), or, alternatively, it can limit the maximum pressure of the boxes placed above a box of type i, so that box i is not deformed and the products inside it are not damaged. In order to avoid empty spaces, or "holes" in the loading pattern, we will consider 100% of vertical stability (i.e., $\alpha_i = 1$ for all i). This same assumption is also present in other studies, such as [44] and [4]. Fig. 6 shows a set of boxes placed above a reference box, pressing it.

Let P_j be the weight of a box of type j, and σ_i the maximum admissible pressure that a box of type i can bear at any point (x', y') of its top face. Note that we assume that any point at the top face of a box of type i can bear the same admissible pressure σ_i. We also assume that the pressure imposed by a box of type j is evenly distributed over the area $l_j w_j$ of its bottom face. Fig. 7 shows some boxes placed above a larger box (the reference box). Consider the sets X, Y and Z defined as the full sets (or as in

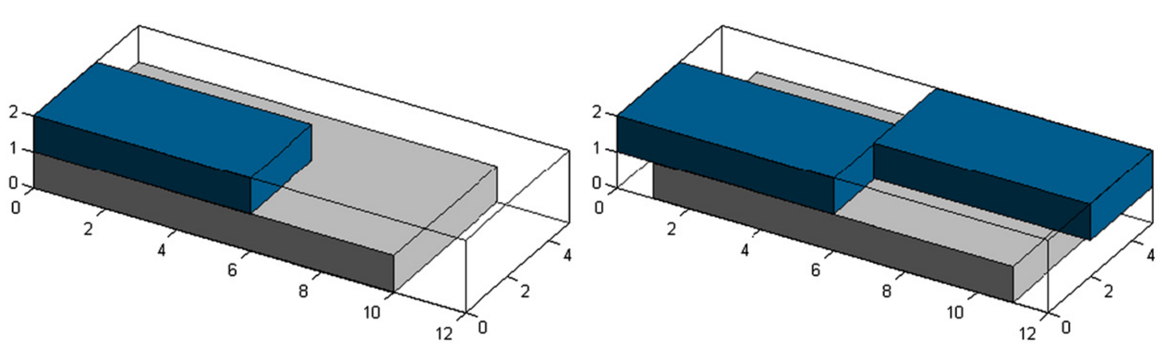

Fig. 5. Loading patterns obtained with constraint (11) with the reduced sets in (1–3), and with the original full sets, respectively.

(1)–(3)). Without loss of generality, the load bearing constraint can be stated as:

$$\sum_{j=1}^{m} \sum_{\{x' \in X_j | x'-l_j+1 \le x' \le x'\}} \sum_{\{y' \in Y_j | y'-w_j+1 \le y' \le y'\}} \sum_{\{z'' \in Z_j | z'+1 \le z' \le H-h_j\}} \left(\frac{P_j}{l_j \cdot w_j}\right) \cdot a_{jx'y'z'} \le \sum_{i=1}^{m} \sum_{\{x \in X_i | x'-l_i+1 \le x \le x'\}} \sum_{\{y \in Y_i | y'-w_i+1 \le y \le y'\}} \sum_{\{z \in Z_i | z'-h_i+1 \le z \le z'\}} \sigma_i \cdot a_{ixyz} \quad x' \in X, \quad y' \in Y, \; z' \in Z, \tag{14}$$

i.e., for each point (x',y',z') of the container, the right side of constraint (14) checks if there is a box of type i that contains this point. If this is true then this right side will be equal to the maximum admissible pressure σ_i that each point at the top face of box of type i can stand (Fig. 7). The left side of constraint (14) corresponds to the sum of the pressure exercised by all boxes of type j stacked above box of type i ($z'' > z'$).

Fig. 8 shows two possible loading patterns for a simple example with a container (L,W,H)=(12,5,2) and three types of boxes: (l_1,w_1,h_1)=(12,5,1) with b_1=2, P_1=60 and σ_1=0.7, (l_2,w_2,h_2)=(5,5,1) with b_2=4, P_2=15 and σ_2=1, and (l_3,w_3,h_3)=(1,5,1) with b_3=3, P_3=2 and σ_3=0.5. Note that both loading patterns occupy the whole container; however, the left pattern of Fig. 8 is infeasible with respect to the load bearing constraint, while the right pattern respects the load bearing requirements of the boxes.

As mentioned, fragility is a particular case of load bearing, in which case no box can be placed above the reference box, because its top face does not bear any kind of pressure. In order to consider this case, it is sufficient to set σ_i=0 for each fragile box of type i on the right hand side of constraint (14).

Note that the *base model* (i.e., without vertical stability and without load bearing), the *base model* with constraints (11) (i.e., with vertical stability), and the *base model* with constraints (11) and (14) (i.e., with vertical stability and with load bearing), consist of $\sum_{i=1}^{m} |X_i| \cdot |Y_i| \cdot |Z_i|$ binary variables. Moreover, the *base model* has $|X| \cdot |Y| \cdot |Z|+m$ constraints, constraints (11) add $\sum_{i=1}^{m} |X_i| \cdot |Y_i| \cdot (|Z_i|-1)$ constraints to the *base model*, and constraints (14) add $|X| \cdot |Y| \cdot |Z|$ constraints to the *base model* with (11).

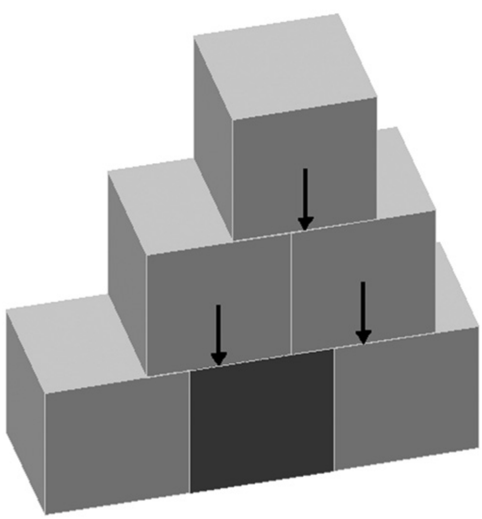

Fig. 6. A set of boxes placed above a reference box.

4. Computational results

Three models of Section 3: the *base model* (i.e., without vertical stability and without load bearing), the *base model* with constraints (11) (i.e., with vertical stability), and the *base model* with constraints (11) and (14) (i.e., with vertical stability and load bearing constraints), were implemented in the modeling language GAMS (version 22.7). Solver CPLEX 11.0 (with default parameters) was used to solve them. All computational tests were performed in a PC Pentium D (3.2 GHz, 2.0 GB) with instances randomly generated, using the following parameters:

- Four types of boxes: m=1 (in this case, the boxes can rotate around all axes), m=5, m=10 and m=20 (in these three cases, the boxes have fixed orientation).
- Boxes dimensions generated in two different ways: (A_m, m=1, 5, 10 and 20) with box dimensions varying between 25% and 75% of the dimensions of the container, i.e., $l_i \in [0.25L, 0.75L]$, $w_i \in [0.25W, 0.75W]$ and $h_i \in [0.25H, 0.75H]$; (B_m, m=1, 5, 10 and 20) with boxes dimensions varying between 10% and 50% of the container dimensions, i.e., $l_i \in [0.10L, 0.50L]$, $w_i \in [0.10W, 0.50W]$ and $h_i \in [0.10H, 0.50H]$. For the sake of simplicity, in all examples we consider cubic containers, i.e., with dimensions L=W=H.

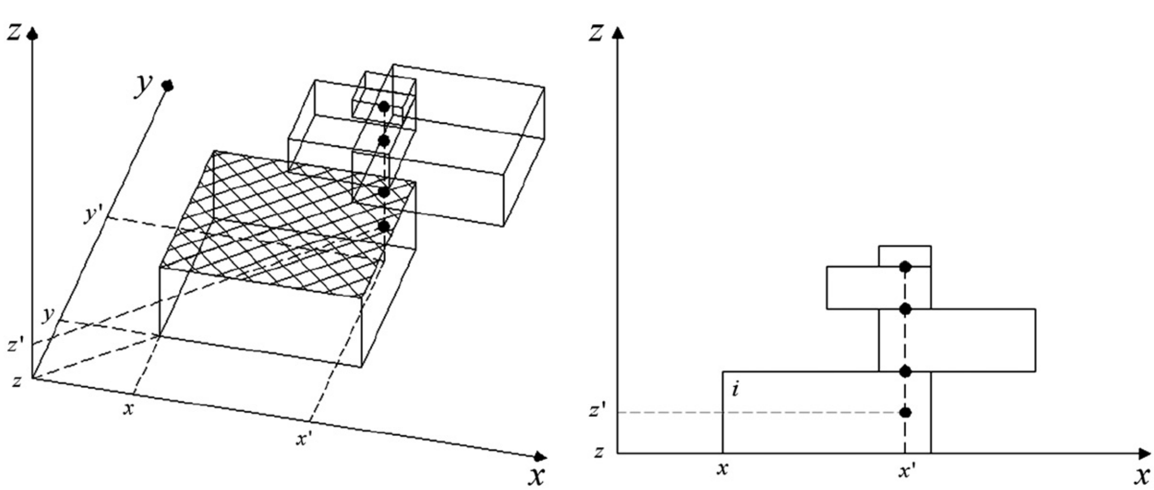

Fig. 7. Set of boxes stacked over the reference box.

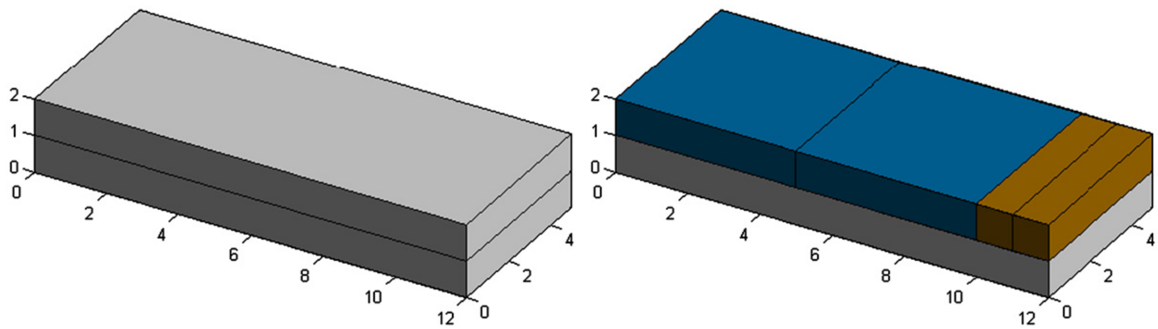

Fig. 8. Infeasible and feasible loading patterns, respectively, with respect to the load bearing constraint.

Table 1
Dimension $L=W=H$ of the containers for the groups of instances generated randomly.

a_m	A_m (25%, 75%)	B_m (10%, 50%)
1	10, 20, 30, 50 and 100	10, 20 and 30
5	10, 20, 30, 50 and 100	10, 20 and 30
10	10, 20, 30, 50 and 100	10, 20 and 30
20	10, 20, 30, 50 and 100	10, 20 and 30

When $m=1$, an additional decision variable for each possible box orientation was defined, resulting in a total of six decision variables, and the *base model*, the *base model with* (11), and the *base model* with (11) and (14) were properly modified to consider these new variables. An alternative to handle this case would be to consider each of the six possible rotations of a box as a different box type, i.e., $m=6$, and to limit the maximum number of boxes loaded in constraints (6). As mentioned in Section 1, a motivation for problems of group A_1 and B_1 (without fixed orientation for the boxes) is that they can be seen as a three-dimensional manufacturer's pallet loading problem where the boxes do not need to be arranged in horizontal layers on the pallet.

Table 1 presents the dimensions $L=W=H$ of the containers, ranging from 10 to 100. The first column shows the number of box types, the second and third columns, respectively, show the box dimensions generated for each category A_m and B_m. In general, group A_m is comprised of easier problems when compared to B_m, since the size of the boxes generated in the first group is, on average, larger than the ones generated in B_m, which in turn leads to a smaller number of normal patterns, on average, in A_m. In group B_m, only instances with dimension up to $L=W=H=30$ were solved because, for different instances, compiling them in GAMS with higher dimension values resulted in memory overflow. Therefore, for each dimension value of the containers, 10 instances were generated, in a total of 320 instances to be solved by each of the models.

Value v_i was defined as the percentage of the container volume used by a box of type i, i.e.: $v_i=[(l_i \cdot w_i \cdot h_i)/(L \cdot W \cdot H)]$, $i=1,\ldots,m$. The amount of boxes available b_i that can be loaded in the container was defined in two different ways. For instances where $m=5$, 10 and 20, the value of b_i was randomly generated by a uniform distribution in the interval: $[1, \lfloor L/l_i \rfloor \cdot \lfloor W/w_i \rfloor \cdot \lfloor H/h_i \rfloor]$, $i=1,\ldots,m$. Note that, with these values for b_i, the KLP becomes constrained with respect to the availability of box type i (in the literature of cutting and packing, the problem is called constrained if $b_i < \lfloor (L \cdot W \cdot H)/(l_i \cdot w_i \cdot h_i) \rfloor$ for some i). For all instances where $m=1$, the value of b_i was defined as: $b_i = \lfloor (L \cdot W \cdot H)/(l_i \cdot w_i \cdot h_i) \rfloor$, $i=1,\ldots,m$, meaning that the problem is not constrained with respect to the availability of boxes.

The weight P_i was defined as the volume of a box of type i, since we assumed, for the sake of simplicity, that all boxes have the same density: $P_i=(l_i \cdot w_i \cdot h_i)$, $i=1,\ldots,m$. The maximum admissible pressure σ_i that a box can bear in any point of its top face was determined in two different ways. For instances where $m=5$, 10 and 20, the value of σ_i was randomly generated by a uniform distribution in the interval: $[0,3h_i]$, $i=1,\ldots,m$. Since we assumed that the boxes have a fixed orientation, and the weight of each box is its own volume, we defined the maximum admissible pressure as being proportional to the height of the box. In the literature, the constant value of 3 is the most commonly used to multiply the value of the height of the box [44]. Note that, if the chosen value of σ_i is 0, then the box is considered fragile and, therefore, cannot have another box placed on top of it. For instances where $m=1$, the value of σ_i was defined in a similar way, but now we have taken into account which face of the box will be used as its supporting base: $[0,3h_i]$ if the supporting base is l_iw_i, $[0,3w_i]$ if the supporting base is l_ih_i, and $[0,3l_i]$ if the supporting base is w_ih_i, $i=1,\ldots,m$. The value of the parameter of vertical stability α_i was set to 1 for all box types i, i.e., the bottom faces of all boxes must be 100% supported by the top faces of one or more boxes placed immediately below them (or by the container's floor). Note that, for the *base model* with (11), condition $\alpha_i=1$ ensures that optimal solutions will not be lost when using normal patterns, and for the *base model* with (11) and (14), condition $\alpha_i=1$ is necessary to avoid "holes" in the loading pattern, as discussed in Section 3.

In order to illustrate the sizes of the models generated using instances of groups A_m and B_m, Table 2 presents the minimum, average and maximum sizes of sets X, Y and Z in (1)–(3), and the minimum, average and maximum number of constraints and binary variables, for each of the eight groups, considering a container with dimensions $L=W=H$ equal to 10. These numbers correspond to the values reported by CPLEX after the pre-processing. Note that the average reduction (in percent) in the number of variables obtained by CPLEX after the pre-processing for each one of the eight groups considered in this table are 74.07, 75.23, 70.63, 69.34, 38.36, 40.80, 39.99 and 38.96.

As can be observed in Table 2, the number of variables and constraints increase significantly with larger values of m (compare, for instance, A_5 and A_{20}) and with smaller sizes of the boxes (compare, for instance, A_5 and B_5). Note that, even for instances where $L=W=H$ are smaller (i.e., equal to 10), the models can involve 1000s of variables and constraints. In the experiments that follow, the computational time spent to solve each model was limited to 1 h (3600 s) and the optimality gaps were computed as:

$$Gap = \frac{(\text{best bound obtained} - \text{best solution obtained})}{(\text{best bound obtained})} \times 100\%.$$

Therefore, four possible cases, with respect to the quality of the solution obtained by GAMS/CPLEX, can occur: (i) optimal solution, with gap equals to zero; (ii) integer solution, with gap greater

Table 2
Number of elements of the normal pattern sets and number of constraints and variables of the models with $L=W=H=10$.

	Stat.	No. normal patterns			No. var.	No. constraints		
		$\lvert X\rvert$	$\lvert Y\rvert$	$\lvert Z\rvert$		Base model	Base model with (11)	Base model with (11) and (14)
A_1	Min.	3.00	3.00	3.00	73.00	29.00	69.00	96.00
	Avg.	**6.10**	**6.10**	**6.10**	**353.20**	**266.80**	**532.60**	**797.40**
	Max.	9.00	9.00	9.00	1345.00	731.00	1843.00	2572.00
A_5	Min.	5.00	6.00	6.00	294.00	246.00	492.00	732.00
	Avg.	**7.50**	**8.10**	**7.60**	**571.80**	**472.40**	**925.60**	**1392.00**
	Max.	9.00	9.00	9.00	1237.00	654.00	1710.00	2358.00
A_{10}	Min.	5.00	8.00	8.00	1327.00	416.00	1643.00	2149.00
	Avg.	**8.50**	**8.90**	**8.80**	**1955.40**	**676.10**	**2288.70**	**2953.80**
	Max.	9.00	9.00	9.00	2398.00	740.00	2760.00	3489.00
A_{20}	Min.	9.00	9.00	9.00	3949.00	750.00	3978.00	4707.00
	Avg.	**9.00**	**9.00**	**9.00**	**4469.90**	**750.00**	**4476.80**	**5205.80**
	Max.	9.00	9.00	9.00	5423.00	750.00	5354.00	6083.00
B_1	Min.	9.00	9.00	9.00	2353.00	731.00	2761.00	3490.00
	Avg.	**9.80**	**9.80**	**9.80**	**3481.00**	**947.80**	**4009.20**	**4955.00**
	Max.	10.00	10.00	10.00	5401.00	1002.00	5842.00	6842.00
B_5	Min.	5.00	10.00	9.00	1251.00	456.00	1529.00	1979.00
	Avg.	**9.00**	**10.00**	**9.80**	**2610.70**	**891.00**	**3186.80**	**4071.80**
	Max.	10.00	10.00	10.00	3412.00	1006.00	4045.00	5045.00
B_{10}	Min.	9.00	10.00	10.00	4889.00	911.00	5206.00	6106.00
	Avg.	**9.80**	**10.00**	**10.00**	**5881.00**	**991.00**	**6181.80**	**7161.80**
	Max.	10.00	10.00	10.00	6628.00	1011.00	6867.00	7867.00
B_{20}	Min.	10.00	10.00	10.00	11448.00	1021.00	11097.00	12097.00
	Avg.	**10.00**	**10.00**	**10.00**	**12208.80**	**1021.00**	**11797.70**	**12797.70**
	Max.	10.00	10.00	10.00	12906.00	1021.00	12420.00	13420.00

Table 3
Results obtained for group A_1.

	Stat.	Base model				Base model with (11)				Base model with (11) and (14)			
		No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)
10	Min.	10	0.000	0.08	64.00	10	0.000	0.08	54.00	10	0.000	0.09	54.00
	Avg.		**0.000**	**312.90**	**79.42**		**0.770**	**361.31**	**77.98**		**0.000**	**1.16**	**75.58**
	Max.		0.000	3126.42	96.00		7.700	3600.17	96.00		0.000	7.67	96.00
20	Min.	10	0.000	0.06	39.20	10	0.000	0.06	39.20	10	0.000	0.13	39.20
	Avg.		**0.000**	**0.20**	**67.35**		**0.000**	**4.09**	**67.35**		**0.000**	**3.74**	**65.30**
	Max.		0.000	0.91	90.00		0.000	38.73	90.00		0.000	31.45	90.00
30	Min.	10	0.000	0.05	37.33	10	0.000	0.03	37.33	10	0.000	0.11	37.33
	Avg.		**0.000**	**0.08**	**59.43**		**0.000**	**0.12**	**56.96**		**0.000**	**0.19**	**56.96**
	Max.		0.000	0.13	80.00		0.000	0.23	80.00		0.000	0.36	80.00
50	Min.	10	0.000	0.06	33.18	10	0.000	0.06	33.18	10	0.000	0.13	33.18
	Avg.		**0.000**	**3.03**	**59.74**		**0.000**	**64.66**	**58.47**		**0.000**	**182.74**	**56.84**
	Max.		0.000	28.91	87.09		0.000	643.39	87.09		0.000	1749.84	70.76
100	Min.	10	0.000	0.06	20.33	10	0.000	0.06	20.33	10	0.000	0.03	20.33
	Avg.		**0.000**	**0.86**	**54.11**		**0.000**	**149.27**	**53.03**		**0.000**	**0.78**	**53.03**
	Max.		0.000	7.59	97.20		0.000	1436.66	86.40		0.000	6.44	86.40

than zero and with CPLEX exceeding the time limit; (iii) no solution, without gap and with CPLEX exceeding the time limit; (iv) insufficient computer memory to compile the model in GAMS, no gap and no relevant information concerning the computational time. The last two cases are represented in the tables by the symbol "–".

The following tables show the minimum, average and maximum values for: the optimality gap (in %), the runtime (in seconds) spent to solve the model, and the fraction of volume (in %) occupied by the boxes loaded in the container. We highlight here that the minimum, average and maximum values, were computed only for instances solved optimally, or for those instances where a suboptimal integer solution was found. Note that the first column of the tables refers to the dimension of the container ($L=W=H$).

4.1. Results of groups A_m

The results obtained with the 200 instances of groups A_1, A_5, A_{10} and A_{20}, for the three models, are presented in Tables 3–6. As expected, due to the method used to generate the instances in A_m, the average fraction of volume occupied increases with m (compare, for instance, the results in Tables 4 and 6 with $m=5$

Table 4
Results obtained for group A_5.

	Stat.	Base model				Base model with (11)				Base model with (11) and (14)			
		No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)
10	Min.	10	0.000	0.13	77.20	10	0.000	0.25	76.00	10	0.000	0.30	47.40
	Avg.		**0.000**	**2.73**	**91.34**		**0.000**	**38.07**	**89.87**		**0.440**	**418.24**	**71.44**
	Max.		0.000	13.42	100.00		0.000	236.91	100.00		4.400	3600.19	93.60
20	Min.	10	0.000	0.16	60.69	10	0.000	0.17	55.53	10	0.000	0.08	52.85
	Avg.		**0.000**	**4.59**	**78.54**		**0.000**	**50.71**	**75.50**		**0.000**	**78.31**	**66.73**
	Max.		0.000	40.53	91.10		0.000	407.53	87.80		0.000	763.80	81.40
30	Min.	10	0.000	0.13	42.93	10	0.000	0.11	42.93	10	0.000	0.03	42.93
	Avg.		**0.000**	**0.22**	**60.90**		**0.000**	**0.25**	**60.87**		**0.000**	**0.28**	**59.62**
	Max.		0.000	0.36	76.18		0.000	0.69	76.18		0.000	0.80	70.84
50	Min.	10	0.000	0.06	51.48	10	0.000	0.06	51.48	10	0.000	0.08	50.01
	Avg.		**0.000**	**1.36**	**70.08**		**0.000**	**164.20**	**67.49**		**0.000**	**14.04**	**64.37**
	Max.		0.000	5.69	84.85		0.000	1601.25	82.60		0.000	114.64	75.79
100	Min.	10	0.000	0.09	48.70	10	0.000	0.08	46.52	10	0.000	0.09	43.77
	Avg.		**0.000**	**0.54**	**68.24**		**0.000**	**0.83**	**64.04**		**0.000**	**1.69**	**59.63**
	Max.		0.000	1.61	85.04		0.000	2.25	85.04		0.000	6.45	85.04

Table 5
Results obtained for group A_{10}.

	Stat.	Base model				Base model with (11)				Base model with (11) and (14)			
		No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)
10	Min.	10	0.000	0.34	98.40	10	0.000	1.25	96.80	10	0.000	5.52	83.20
	Avg.		**0.000**	**191.33**	**99.54**		**0.190**	**879.51**	**99.38**		**4.180**	**1600.76**	**94.02**
	Max.		0.000	1505.83	100.00		1.300	3600.23	100.00		16.600	3600.63	100.00
20	Min.	10	0.000	1.34	76.45	10	0.000	4.14	70.76	10	0.000	3.27	64.62
	Avg.		**0.890**	**857.87**	**89.95**		**2.970**	**1607.21**	**85.82**		**4.850**	**1687.84**	**78.38**
	Max.		5.100	3600.75	94.30		16.500	3600.80	94.30		18.200	3601.63	94.30
30	Min.	10	0.000	2.08	70.06	10	0.000	3.00	63.83	10	0.000	11.50	58.67
	Avg.		**0.470**	**605.15**	**83.05**		**1.840**	**1036.90**	**78.88**		**2.050**	**855.64**	**76.57**
	Max.		4.700	3601.03	93.56		14.800	3603.77	93.56		13.900	3603.02	86.59
50	Min.	9	0.000	20.42	77.36	9	0.000	43.41	68.14	7	0.000	18.28	60.45
	Avg.		**2.800**	**2236.48**	**82.02**		**5.767**	**2821.08**	**77.05**		**4.786**	**2138.93**	**68.25**
	Max.		10.800	3607.25	85.85		14.200	3606.31	84.82		14.900	3605.75	73.89
100	Min.	4	0.000	150.17	65.70	9	0.000	59.02	59.21	7	0.000	79.63	58.87
	Avg.		**0.925**	**1139.19**	**78.12**		**4.222**	**2142.94**	**78.69**		**2.857**	**2121.36**	**70.93**
	Max.		3.700	3604.69	88.37		16.700	3608.30	93.09		13.400	3646.27	81.27

Table 6
Results obtained for group A_{20}.

	Stat.	Base model				Base model with (11)				Base model with (11) and (14)			
		No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)	No ex.	Gap (%)	Time (s)	Vol. (%)
10	Min.	10	0.000	6.41	99.60	10	0.000	57.45	99.60	10	0.000	82.14	93.40
	Avg.		**0.000**	**10.40**	**99.96**		**0.000**	**303.92**	**99.96**		**1.070**	**2149.72**	**98.92**
	Max.		0.000	15.02	100.00		0.000	2355.89	100.00		6.600	3601.16	100.00
20	Min.	10	0.000	972.91	90.71	10	0.000	2830.84	80.77	10	0.800	3601.66	80.50
	Avg.		**2.080**	**2896.56**	**95.03**		**5.960**	**3524.60**	**91.57**		**10.240**	**3603.34**	**86.44**
	Max.		4.200	3603.55	97.69		15.800	3602.67	97.20		17.200	3605.22	93.75
30	Min.	7	0.000	518.92	86.42	7	1.300	3602.16	80.01	4	0.000	1276.27	80.59
	Avg.		**1.829**	**2215.66**	**91.85**		**6.729**	**3604.20**	**87.66**		**1.375**	**2280.82**	**87.19**
	Max.		5.500	3605.64	97.21		11.000	3606.03	92.89		5.500	3610.33	92.92
50	Min.	0	–	–	–	0	–	–	–	0	–	–	–
	Avg.		**–**	**–**	**–**		**–**	**–**	**–**		**–**	**–**	**–**
	Max.		–	–	–		–	–	–		–	–	–
100	Min.	0	–	–	–	0	–	–	–	0	–	–	–
	Avg.		**–**	**–**	**–**		**–**	**–**	**–**		**–**	**–**	**–**
	Max.		–	–	–		–	–	–		–	–	–

Table 7
Summary of the results for group A_m.

	Base model			Base model with (11)			Base model with (11) and (14)		
	No exp.	No integer sol.	No optimal sol.	No exp.	No integer sol.	No optimal sol.	No exp.	No integer sol.	No optimal sol.
A_1	50	50	50	50	50	49	50	50	50
A_5	50	50	50	50	50	50	50	50	49
A_{10}	50	43	35	50	48	30	50	44	29
A_{20}	50	27	17	50	27	11	50	24	9
Total	200	170	152	200	175	140	200	168	137

Table 8
Summary of results from group B_m.

	Base model			Base model with (11)			Base model with (11) and (14)		
	No exp.	No integer sol.	No optimal sol.	No exp.	No integer sol.	No optimal sol.	No exp.	No integer sol.	No optimal sol.
B_1	30	27	24	30	24	13	30	25	16
B_5	30	22	12	30	17	8	30	16	6
B_{10}	30	10	10	30	10	9	30	8	2
B_{20}	30	10	10	30	7	7	30	1	0
Total	120	69	56	120	58	37	120	50	24

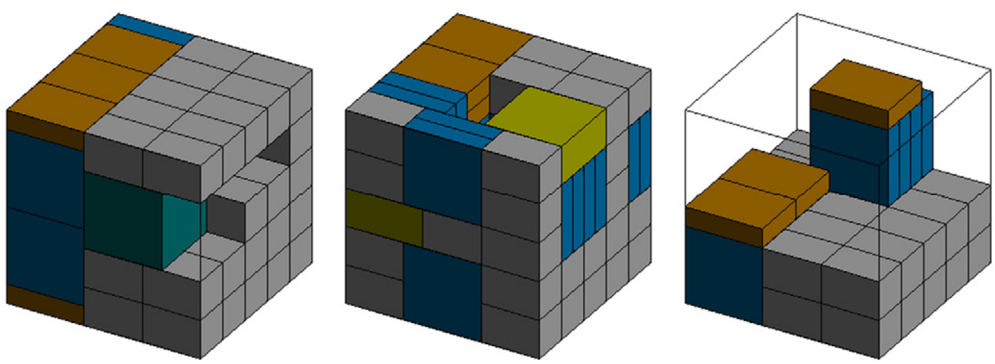

Fig. 9. Loading patterns obtained when solving the same instance with the three models.

and 20, respectively). On the other hand, for each group A_m, the average fraction of the volume occupied decreases when the dimensions $L=W=H$ of the containers increase, as expected. Note also that the (optimal) solution yielded by the model with stability and load bearing can be different from the (optimal) solution yielded by the *base model*, showing that the last may not satisfy the stability and load bearing constraints.

To illustrate the quality of the linear relaxation bound of the models, we solved the base model with (11) using the 50 instances of group A_5 (Table 4). These linear relaxation bounds were computed in the same way as the optimality gaps, only changing in the latter the expression "best bound obtained" to "best linear relaxation bound obtained". The average linear relaxation gap obtained from these 50 instances was 6.706% (minimum of 0.000% and maximum of 27.216%).

Table 7 summarizes the results of the instances tested for group A_m. For each model solved, this table presents the number of instances tested, the number of instances where at least one integer solution was obtained, and the number of instances solved optimally. It is possible to observe that, for the model without vertical stability, in 85.00% of the instances (170 out of 200), an integer solution was obtained, and in 89.41% of them (152 out of 170), an optimal solution was found. For the models with vertical stability and vertical stability with load bearing, these numbers resulted in 87.50% and 80.00%, and in 75.00% and 86.67%, respectively. It is worth noting that GAMS/CPLEX starts to have difficulties in solving the models in groups A_{10} and A_{20}, due to the time limit and computational memory (items (iii) and (iv) mentioned previously).

4.2. Results of groups B_m

The results obtained with the 120 instances of groups B_1, B_5, B_{10} and B_{20} for the three models are summarized in Table 8. Some remarks concerning the results presented in Table 7 also apply to Table 8. Note that GAMS/CPLEX has difficulties in solving the instances from groups B_{10} and B_{20}, even for relatively small sized instances, for example, when $L=W=H$ is equal to 20 or 30.

Fig. 9 compares the loading pattern of an instance solved using the *base model*, the *base model* with (11) and the *base model* with (11) and (14), respectively. The *base model* loads 65 boxes and occupies 95.20% of the container volume. Note that, in this loading pattern, there are boxes that are only partially supported, or are "floating" (in mid-air) inside of the container, i.e., this loading is not 100% stable. The *base model* with (11) presents a solution with 68 boxes, which occupies the same 95.20% of the volume of the container. It is interesting to note that the empty spaces concentrated at the top of the container, and the loading is 100% stable. The solution obtained by the *base model* with (11) and (14), due to the restriction of the maximum admissible pressure, loads only 35 boxes and occupies 50.00% of the volume of the container.

The computational results presented in this section show the limitations of the software GAMS/CPLEX (with default parameters) to solve models with randomly generated instances of a moderate size. The results lead us to believe that this approach will have difficulties in dealing with realistic container loading problems that involve a great number of possible positions to arrange the boxes inside the container (for instance, the examples presented in [20,5]). Nevertheless, it is worth noting that the models solved in this section have 1000s of variables and constraints and are far from being considered easy to solve.

5. Conclusion

In this paper, we present 0-1 integer linear programming models for the problem of loading rectangular boxes inside objects (containers, trucks, railway cars or pallets), considering constraints of cargo stability and load bearing. The models can be further extended to consider other practical considerations, some of them mentioned in Section 2. Moreover, the models can also be adapted to deal with three-dimensional strip packing (SPP) and bin packing (BPP) variants. Computational tests were performed with 320 instances randomly generated and solved by GAMS/CPLEX. The results show that the models are consistent and properly represent the practical situations treated, although this approach (in its current version) is limited to solving optimally only problems of a moderate size, i.e., where the number of possible positions to arrange the boxes in the container is relatively small.

The proposed models can be useful to motivate future research exploring decomposition methods, relaxation methods, heuristics, among others, in order to solve more realistic container loading problems. Other interesting topics for future research are: (i) to extend the models to consider other practical considerations beyond cargo stability and load bearing, (ii) to perform experiments to find out the best combination of parameter's values of CPLEX (for the sake of simplicity, all parameters of CPLEX were set to their default values), (iii) to explore the use of constraints to reduce symmetry in the loading patterns and to investigate if the normal sets described in (1–3) and applied to the models, can be replaced, without loss of generality, by sets of smaller cardinality, such as the raster points sets [45,3], in order to reduce the size of the models, (iv) to integrate the models in coupled vehicle routing and container loading models with multi-drop constraints.

Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments and suggestions. This research was partially supported by FAPESP (Grants 06/03496-3, 07/00209-6 and 07/01299-9) and CNPq (Grant 522973/95-7).

References

[1] Beasley JE. An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research 1985;33(1):49–64.
[2] Beasley JE. A population heuristic for constrained two-dimensional non-guillotine cutting. European Journal of Operational Research 2004;156(3):601–27.
[3] Birgin EG, Lobato RD, Morabito R. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. Journal of the Operational Research Society 2010;61(2):306–20.
[4] Bischoff EE. Three-dimensional packing of items with limited load bearing strength. European Journal of Operational Research 2006;168(3):952–66.
[5] Bischoff EE, Ratcliff MSW. Issues in the development of approaches to container loading. Omega 1995;23(4):377–90.
[6] Bortfeldt A, Gehring H. A hybrid genetic algorithm for the container loading problem. European Journal of Operational Research 2001;131(1):143–61.
[7] Bortfeldt A, Gehring H, Mack D. A parallel tabu search algorithm for solving the container loading problem. Parallel Computing 2003;29(5):641–62.
[8] Bortfeldt A, Mack D. A heuristic for the three-dimensional strip packing problem. European Journal of Operational Research 2007;183(3):1267–79.
[9] Chen CS, Lee SM, Shen QS. An analytical model for the container loading problem. European Journal of Operational Research 1995;80(1):68–76.
[10] Chien CF, Deng JF. A container packing support system for determining and visualizing container packing patterns. Decision Support Systems 2004;37(1):23–34.
[11] Christensen SG, Rousøe DM. Container loading with multi-drop constraints. International Transactions in Operational Research 2009;16(6):727–43.
[12] Christofides N, Whitlock C. An algorithm for two-dimensional cutting problems. Operations Research 1977;25(1):30–44.
[13] Davies AP, Bischoff EE. Weight distribution considerations in container loading. European Journal of Operational Research 1999;114(3):509–27.
[14] Dyckhoff H. A typology of cutting and packing problems. European Journal of Operational Research 1990;44(2):145–59.
[15] Eley M. Solving container loading problems by block arrangement. European Journal of Operational Research 2002;141(2):393–409.
[16] Eley M. A bottleneck assignment approach to the multiple container loading problem. OR Spectrum 2003;25(1):45–60.
[17] Gehring H, Bortfeldt A. A parallel genetic algorithm for solving the container loading problem. International Transactions in Operational Research 2002;9(4):497–511.
[18] Gendreau M, Iori M, Laporte G, Martello S. A tabu search algorithm for a routing and container loading problem. Transportation Science 2006;40(3):342–50.
[19] George JA. A method for solving container packing for a single size of box. Journal of the Operational Research Society 1992;43(4):307–12.
[20] George JA, Robinson DF. A heuristic for packing boxes into a container. Computers and Operations Research 1980;7(3):147–56.
[21] Gilmore PC, Gomory RE. Multistage cutting stock problems of two and more dimensions. Operations Research 1965;13(1):94–120.
[22] Hadjiconstantinou E, Chistofides N. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. European Journal of Operational Research 1995;83(1):39–56.
[23] Hifi M. Approximate algorithms for the container loading problem. International Transactions in Operational Research 2002;9(6):747–74.
[24] Hodgson TJ. A combined approach to the pallet loading problem. IIE Transactions 1982;14(3):175–82.
[25] Huang W, He K. A caving degree approach for the single container loading problem. European Journal of Operational Research 2009;196(1):93–101.
[26] Jin Z, Ohno K, Du J. An efficient approach for the three-dimensional container packing problem with practical constraints. Asia-Pacific Journal of Operational Research 2004;21(3):279–95.
[27] Junqueira L, Morabito R, Yamashita DS. Modelos de otimização para problemas de carregamento de contêineres com considerações de estabilidade e de empilhamento. Pesquisa Operacional 2010;30(1):73–98.
[28] Lai KK, Xue J, Xu B. Container packing in a multi-customer delivering operation. Computers and Industrial Engineering 1998;35(1–2):323–6.
[29] Li HL, Tsai JF, Hu NZ. A distributed global optimization method for packing problems. Journal of the Operational Research Society 2003;54(4):419–25.
[30] Lim A, Rodrigues B, Wang Y. 3-D container packing heuristics. Applied Intelligence 2005;22(2):125–34.
[31] Lin JL, Chang CH, Yang JY. A study of optimal system for multiple-constraint multiple-container packing problems. In: Proceedings of the 19th international conference on industrial, engineering and other applications of applied intelligent systems, vol. 4031, Annecy, 2006. p. 1200–10.
[32] Lins L, Lins S, Morabito R. An *n*-tet graph approach for non-guillotine packing of *n*-dimensional boxes into an *n*-container. European Journal of Operational Research 2002;141(2):421–39.
[33] Lins L, Lins S, Morabito R. An *L*-approach for packing (l,w)-rectangles into rectangular and *L*-shaped pieces. Journal of the Operational Research Society 2003;54(7):777–89.
[34] Mack D, Bortfeldt A, Gehring H. A parallel hybrid local search algorithm for the container loading problem. International Transactions in Operational Research 2004;11(5):511–33.
[35] Martello S, Pisinger D, Vigo D. The three-dimensional bin packing problem. Operations Research 2000;48(2):256–67.
[36] Martins GHA. Packing in two and three dimensions. PhD thesis, Naval Postgraduate School, Monterey, California; 2003.
[37] Miyazawa FK, Wakabayashi Y. Approximation algorithms for the orthogonal *Z*-oriented three-dimensional packing problem. SIAM Journal on Computing 1999;29(3):1008–29.
[38] Morabito R, Arenales M. An And/Or-graph approach to the container loading problem. International Transactions in Operational Research 1994;1(1):59–73.
[39] Moura A, Oliveira JF. A GRASP approach to the container-loading problem. IEEE Intelligent Systems 2005;4(20):50–7.

[40] Moura A, Oliveira JF. An integrated approach to the vehicle routing and container loading problems. OR Spectrum 2009;31(4):775–800.
[41] Parreño F, Alvarez-Valdes R, Oliveira JF, Tamarit JM. A maximal-space algorithm for the container loading problem. INFORMS Journal on Computing 2008;20(3):412–22.
[42] Parreño F, Alvarez-Valdes R, Oliveira JF, Tamarit JM. Neighborhood structures for the container loading problem: a VNS implementation. Journal of Heuristics 2010;16(1):1–22.
[43] Pisinger D. Heuristics for the container loading problem. European Journal of Operational Research 2002;141(2):382–92.
[44] Ratcliff MSW, Bischoff EE. Allowing for weight considerations in container loading. OR Spektrum 1998;20(1):65–71.
[45] Scheithauer G, Terno J, Riehme J, Sommerweiss U. A new heuristic approach for solving the multi-pallet packing problem. Technical Report, MATH-NM-03-1996, Technische Universität Dresden, Dresden; 1996.
[46] Silva JLC, Soma NY, Maculan N. A greedy search for the three-dimensional bin packing problem: the packing static stability case. International Transactions in Operational Research 2003;10(2):141–53.
[47] Terno J, Scheithauer G, Sommerweiss U, Riehme J. An efficient approach for the multi-pallet loading problem. European Journal of Operational Research 2000;123(2):372–81.
[48] Tsai RD, Malstrom EM, Kuo W. Three dimensional palletization of mixed box sizes. IIE Transactions 1993;25(4):64–75.
[49] Wang Z, Li KW, Levy JK. A heuristic for the container loading problem: a tertiary-tree-based dynamic space decomposition approach. European Journal of Operational Research 2008;191(1):86–99.
[50] Wäscher G, Haussner H, Schumann H. An improved typology of cutting and packing problems. European Journal of Operational Research 2007;183(3): 1109–30.
[51] Yeung LHW, Tang WKS. A hybrid genetic approach for container loading in logistics industry. IEEE Transactions on Industrial Electronics 2005;52(2):617–27.